



Ciclo while

Transcripción

[00:00] Sean bienvenidos a una clase más de su curso Primeros pasos en Java. Ya sabemos declarar distintos tipos de variables como enteros, booleanos, ya conocemos expresiones booleanas que son usadas para condicionales if, sabemos usar else. Y una cosa más que necesitamos aprender para ya tener un nivel básico de Java es el uso de ciclos.

[00:26] ¿Qué es un ciclo? Básicamente es ejecutar cierta parte de código mientras se cumpla una condición o durante una condición verdadera. Vamos a ver esto más claro en código, vamos a crear una nueva clase y vamos a poner de ejemplo ciclos. Le damos a finish, creamos nuestra clase. Como es de costumbre creamos nuestro método Main, con doble espacio. Perfecto.

[00:58] A este nivel ya estamos muy rápido creando el método Main, haciendo nuestros sysouts. Perfecto. Entonces, pongamos de ejemplo que yo necesito imprimir todos los números del 1 al 10. Si yo tuviera que imprimir todos los números del 1 a 10 es un candidato ideal para yo ejecutar un ciclo.

[01:24] Un ciclo básicamente es una tarea repetitiva, como ya les dije, con el cual por ejemplo yo voy a decir que esta variable contador que yo la inicializo en cero, yo quiero decirle que por cada ciclo aumente 0 en 1, e imprimir ese valor dentro del ciclo. Para lograr esto necesitamos una palabra reservada de Java, un operador de Java, llamado while.

[01:58] While es el encargado de ejecutar mi ciclo siempre que la condición que está dentro de sus paréntesis sea verdadera. Vamos a entrar un poco más al

detalle con esto. Creamos las llaves, perfecto. Al igual que la estructura de if o else, while igual también tiene su propio scope o alcance.

[02:20] Lo que él evalúa aquí dentro es una expresión booleana. Por ejemplo, si yo defino while, true, él compila muy bien. ¿Por qué? Porque la expresión booleana de aquí dentro da como resultado true. ¿Cuál es el problema con esto? Que yo voy a tener la expresión de código aquí ejecutándose infinitamente.

[02:41] While es como si nosotros dijéramos "mientras que", y entre los paréntesis condición, sea verdadero, mientras que, condición, sea verdadero, entonces, "ejecuta esto". Es tan simple como esto. Entonces, si nosotros quisiéramos escribir todos los números del 1 al 10, podría ser algo tan sencillo como: mientras contador sea menor que 10, entonces ejecuta esto.

[03:17] Podría ser entonces un sysout, "Ctrl + espacio", imprimir contador. Perfecto. ¿Cuál es el problema con este código? Que hasta este momento contador siempre va a ser 0, en ningún momento estamos alterando el valor actual de contador. Para esto, vamos a decirle entonces que contador ahora va a ser igual a contador más 1.

[03:47] ¿Con esto qué queremos decir? Que a esta misma variable contador le vas a asignar un nuevo valor que va a ser el mismo incrementado en 1. Guardamos, ejecutamos. Y vemos en efecto que él ha impreso todos los números del 1 al 9. ¿Por qué? ¿Por qué del 1 al 9? Porque simplemente cuando él llega a 9 y él incrementa aquí a 10. Él regresa nuevamente acá porque es cíclico, recuerden que son ciclos.

[04:20] Entonces, él regresa nuevamente a while, y mientras que 10 es menor que 10, eso es falso. Entonces al no cumplirse esta condición, él sale de este ciclo, regresa aquí y no hace nada más. Si nosotros queremos tener en cuenta el número 10 también, ya sabemos, menor o igual, nuevamente. Guardamos, ejecutamos, y ahora sí tenemos todos los números del 1 al 10.

[04:46] Ahora, ¿existe una forma diferente de hacer esto de aquí? Sí. ¿Por qué? Porque incluso en algunos casos una expresión así podría ser considerada anti-patrón. ¿Por qué? Porque yo estoy asignando a esta misma variable el valor de la misma variable más 1 cuando tranquilamente si yo necesito agregar, incrementar el valor de la misma variable en una cierta cantidad, yo puedo hacer que contador sea más igual 1. Voy a comentar esto.

[05:28] Entonces con esta expresión yo le estoy diciendo que a contador cada vez le sume 1. Aquí podría tranquilamente ser 1 o sumarle de 2 en 2 o sumarle de 3 en 3. Es válido. Esta expresión es para yo decirle "increméntame por esta cantidad la misma variable", entonces es un código mucho más entendible que este de aquí porque es muy redundante.

[05:53] Ahora, si yo necesito incrementar de 1 en 1, como estamos haciendo ahora, existe otro tipo de expresión que podemos usar, que es simplemente contador ++. Si ustedes ya han tenido experiencia trabajando con C o lenguajes basados en C, van a ver que esto es muy parecido. Esto también.

[06:15] ¿Con contador ++ qué le estamos diciendo? Increméntame en 1 el contador. Tienen casos en los que de repente ustedes pueden haber visto que es ++ contador. Tiene el mismo efecto pero la diferencia no tiene mucho caso para discutirla, recuerden que estamos en los primeros pasos. Vamos a guardar, vamos a ejecutar y vemos que el resultado obtenido en la consola es el mismo.